



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/786,004	02/26/2004	Russell Norman Owen	13210-16	3548
1059 7590 06/09/2008 BERESKIN AND PARR 40 KING STREET WEST BOX 401 TORONTO, ON M5H 3Y2 CANADA				
EXAMINER				
GUPTA, MUKTESH G				
ART UNIT		PAPER NUMBER		
2144				
MAIL DATE		DELIVERY MODE		
06/09/2008		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

### Office Action Summary

**Application No.**

10/786,004

**Applicant(s)**

OWEN ET AL.

**Examiner**

Muktesh G. Gupta

**Art Unit**

2144

**Period for Reply** -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 20 March 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-50 is/are pending in the application.
- 4a) Of the above claim(s) 4, 5, 27 and 28 is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-3, 6-26 and 29-50 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SI/08)
- 4) ☐ Interview Summary (PTO-413)
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_
- Paper No(s)/Mail Date \_\_\_\_\_

### DETAILED ACTION

1. **Claims 1, 6-9, 12, 24, 29-32, 38, and 50** are amended.

**Claims 4-5, and 27-28** are cancelled.

**Claims 1-3, 6-26, and 29- 50** have been examined on merits and are pending in this application.

### *Information Disclosure Statement*

2. Information disclosure statement (IDS) submitted on 03/20/2008 being considered by the examiner and initialed and dated copy is attached to this office action.

### *Response to Amendment*

3. Applicant's amendment filed on 03/20/2008 necessitated a new ground(s) of rejection presented in this office action. Applicant's arguments are deemed moot in view of the following new grounds of rejection as explained here below, necessitated by Applicant's substantial amendment (i.e., representing the plurality of web service description elements as nodes in a graph, re-ordering the nodes into a tree data structure so that said symbolic references are resolved in a forward direction, and creating said at least one accelerator output file from said tree data structure.) to the claims which significantly affected the scope thereof.

Applicant's arguments with respect to **Claims 1-50** have been considered but are moot in view of the new ground(s) of rejection.

***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

**4. Claims 1-3, 6-26, and 29- 50** are rejected under 35 U.S.C. 102(e) as being anticipated by US Patent Application Publication No. 20050044197 to Lai, Ray Y., (hereinafter “Lai”).

***As to Claims 1, 24 and 50, Lai anticipates method, web services program, stored on computer-readable medium, of processing a web service description so that said web service description is adapted for use with a mobile device, said web service description comprising a plurality of web service description elements, wherein said method is performed at a computing device remotely coupled to the mobile device, said method comprising the steps of: receiving a web service description, wherein said web service description defines an interface to a web service*** (as stated in par. 0412, lines 1-10, par. 0080, lines 1-2 Service Requesters (or consumers) may be accessing the business services from a variety of mobile devices or a browser. This belongs to the

consumer domain. A client may use a phone to inquire about an account balance, where all other architecture components are part of the Service Provider domain, where the relevant Web Services components may process the balance inquiry and perform transcoding for different client devices wherever necessary. A client may receive an account balance on a PDA, WAP phone, or another device based on a personalization profile. FIG. 17 illustrates an exemplary Web Services architecture showing the Consumer Domain and Service Provider Domain which is remote to the mobile device and where all the processing of requested web services is done):

*creating at least one accelerator output file from said web service description, said creating comprising optimizing said web service description for said mobile device, wherein said at least one accelerator output file comprises an optimized web service description is adapted for processing by said mobile device* (as stated in par. 0213, lines 1, par. 0536, lines 6-8, par. 0167, lines 1-2, par. 1330, lines 1-10, The Service Requester is a consumer of business services. This may be the end-user (as in Business-to-Consumer) or server (as in Business-to-Business scenario). The Service Requester finds the business services from the Service Registry via a Service Proxy (such as an Apache SOAP server). Upon a successful search, the Service Registry, which may be provided by the same Service Provider or by a public Service Registry node, fetches the appropriate service description (for example, WSDL) and returns the service end-points (that is where the business service is located) to the Service Requester. Then the Service Requester can "bind" the business service to the actual service endpoint or location or mobile device. Dynamic service look-up may be

beneficial to managing business services within a large user community, such as millions of content providers and mobile phone subscribers, where business services and content are typically added or modified in real-time. FIG. 106 illustrates exemplary Use Cases for an exemplary FX Spot Rate Quote Web Service according to one embodiment. An objective is to use a simplified Foreign Exchange (FX) Spot Rate Quote Service to illustrate how to wrap a remote FX Quote Service using a mixture of XML messaging (XML document using JAXM) and Remote Procedure Call (JAX-RPC) with secure SOAP Message Service (WS-Security) and Security Assertion Markup Language (SAML) protocol for Single Sign-on purpose. How it works is that, given a Sell/Buy currency pair (for example, Sell US Dollars and Buy Euro Dollars), the FX Quote Service will provide a quote based on the average daily exchange rate. Client will receive quote on his mobile device and can execute from his mobile, for buying or selling stock);

*and transmitting said at least one accelerator output file to said mobile device, wherein said at least one accelerator output file facilitates invocations of said web service by said mobile device* (as stated in par. 0413, lines 1-5, par. 0536, lines 6-8, par. 1340, lines 1-2, par. 1341, lines 1-2, par. 1342, lines 1-8, FIG. 18 illustrates an exemplary detailed Web Services architecture. In the Service Delivery component, of the Service Provider Domain, there is a controller servlet that can handle service requests from the Service Requester's mobile devices or browser. Business data returned from the RPC router servlet may be captured by the service proxy in XML format. The presentation manager may reformat the data and transcoding into HTML, or

WML using XSL if applicable. This enables the Service Requester to view in a format that is displayable on any mobile device or browser. The exemplary system preferably allows users to log in once to access multiple partner services. System preferably validates the user credential against the user access list of the Application Server. In other words, users enter login ID and password at the front page then can access a number of partner Web sites. The exemplary system preferably allows users to enter the Sell and Buy Currency codes to request an indicative FX Spot Rate Quote. This is preferably an online request for a quote, where the exemplary system preferably retrieves the Spot Rate from a remote FX Quote Server);

wherein said optimizing comprises resolving symbolic references in said web service description such that said at least one accelerator output file is parseable by said mobile device in one pass (as stated in par. 0546, lines 1-12, par. 0549, lines 1-8, par. 1158, lines 1-12, par. 0585, lines 1-4, par. 0586, lines 1-4, par. 0587, lines 1-4 and par. 0588, lines 1-4, par. 1459, lines 1-3, FIG. 26 is a SOAP cache sequence diagram that illustrates some details of the SOAP cache Use Cases previously described. The SOAP client may need to look up a service endpoint URL to invoke a remote business service. The SOAP client checks whether the nature of the remote business service supports cached business service information. An indicative foreign exchange rate inquiry is an example of using SOAP cache. The SOAP client may then access the SOAP cache, which may be implemented as a hash table stored in the physical memory of the SOAP client. If the cached data is available and is still timely, the SOAP client may retrieve the business service information from the SOAP cache. Predefined

transaction types may cache transaction contents and results. Cached data may be stored in serialized data objects, databases, XML cache (edge products), and LDAP with time stamp, version, and transaction type/ID. Cache data (read-only) can be represented in XML structure and can be parsed/retrieved via DOM, SAX, JDOM, or Xpath and output is presented to client mobile device. This Web Services pattern may be useful, for example, for portal integration with multiple Service Providers or content providers doing processing at the back-end. Traditional point-to-point partner integration typically requires complex technical customization for each partner's back-end application infrastructure. Using Web Services technology may preferably lower the cost of partner integration because SOAP messaging may be relatively flexible and less costly to implement. It can also accommodate different data formats and platform connectivity. Business information can easily be aggregated from multiple sources to create user-stickiness for the portal service with single front-end. ProfileServlet parses the SOAP request using the JAXM/TSIK Message Provider for optimizing and resolving symbolic references for single sign-on by web service requester using mobile device. Web service description is optimized with administering Multiple Servlet Engines, different configurations for different servlet engines in the server.xml file, different services and resources mapping. Vertical scaling optimizes existing computing capacity by running multiple instances of the SOAP servlet engines. Multiple servlet engines support application partitioning to preferably provide better performance and scalability);

and wherein said resolving comprises representing the plurality of web service description elements as nodes in a graph, re-ordering the nodes into a tree data



structure so that said symbolic references are resolved in a forward direction, and creating said at least one accelerator output file from said tree data structure (as stated in par. 0028-0034, lines 1-5, par. 1484, lines 1-12, par. 0179, lines 1-2, JAXP is a lightweight Java API library for parsing and transforming XML documents. It is a high-level wrapper for different parsers; it can use Xerces or Crimson as the underlying parser. It allows parsing of an XML document using: Event-driven (SAX 2.0). Tree-based (DOM Level 2). XML documents transformation XML to XML/other data format using XSL/XSLT. Rendering to PDF or graphics using Cocoon. The exemplary deployment architecture in FIG. 117 shows a possible deployment when the exemplary system is extended and integrated in a typical financial institution's Enterprise Architecture, where there may be different delivery channels supporting, for example, SMS, WAP phone, Fax and email, and interfaces with the external parties, such as the Stock Exchange and Market Data feed providers. The integration of the Web Services components may be handled either by a public or private Service Registry and a SOAP RPC Router. With a Web Services implementation, the XML data transformation may be implemented by JAXP and XSLT, and the Exchange Gateway is implemented using SOAP messaging and/or SOAP-JMS integration. FIG. 118 is a flowchart of a method for generating a vendor-independent Web Service architecture including a plurality of heterogeneous components in accordance with one or more design patterns according to one embodiment).

***As to Claims 2, and 25, Lai anticipates method, web services program, of claims 1, and 24, wherein said web service description is in Web Service Description Language (as stated in par. 0262, lines 4-9, Web Services technology uses the Web Services Description Language (WSDL) in the XML structure to define the interfaces, network connection, and service end-points).***

***As to Claims 3, and 26, Lai anticipates method, web services program, of claims 1, and 24, wherein invocations of the web service by the mobile device are performed in accordance with a Simple Object Access Protocol (as stated in par. 0265, lines 1-4, architecture framework of Web Service is based on open standards technology, J2EETM, XML, SOAP (Simple Object Access Protocol) and UDDI).***

***As to Claims 6, and 29, Lai anticipates method, web services program, of claims 1, and 24, wherein each of a subset of said plurality of web service description elements is associated with transport protocols not supported by said mobile device, and wherein said optimizing further comprises identifying said subset, and excluding said subset from said at least one accelerator output file created at said creating step (as stated in par. 0972, lines 5-9, Web service technology, middleware integration options use SOAP over HTTPS as the transport which decouples (excludes) the middleware from the data transport integration, making interoperability easier (optimized)).***

**As to Claims 7, and 30,** *Lai anticipates method, web services program, of claims 1, and 24, wherein said optimizing further comprises modifying one or more names associated with each of one or more web service description elements (as stated in par. 0865, lines 1-9, Java transcoding tools allow platform environment parameters, dataset names to be changed (modifying) intelligently to the new target environment).*

**As to Claims 8, and 31,** *Lai anticipates method, web services program, of claims 1, and 24, further comprising validating said at least one accelerator output file (as stated in par. 0331, lines 1-3 and par. 0332, lines 1-3, Web services technology Compose or transform XML documents, validate their well-formed ness using SAX or DOM with XML Schema and if necessary, transform one format to another format or XML variant or render XML documents into different mobile devices or multi-channel).*

**As to Claim 9,** *Lai anticipates method of claim 1, further comprising processing said at least one accelerator output file by identifying web service description elements that define inputs to said web service, a destination, and a format for said inputs from said optimized web service description (as stated in par. 0224, lines 1-6, web services technology is easy and quick to deploy and integration framework support different protocols and message formats, including a variety of industry standards and platforms which are identified, integrated and delivered in minimal time frame).*

***As to Claim 10,*** *Lai anticipates method of claim 9, further comprising invoking said web service by transmitting input data to said destination in said format (as stated in par. 1402, lines 1-16, Open Standards messaging protocols are used for the interaction between different components of web services, Clients, Control Servlet, Reference Data, Service Provider and the Registry Server. Control Servlet handles Presentation-Tier requests, acts as a SOAP client to look up the web services dynamically using JAXR, retrieves reference data using JAXM and invokes remote web services using JAX-RPC).*

***As to Claim 11,*** *Lai anticipates method of claim 10, further comprising receiving output data from said web service in response to said invoking step (as stated in par. 0415, lines 1-6, business data returned from the RPC router servlet is captured by the service proxy in XML format. The presentation manager reformats the data and transcode into HTML, or WML using XSL if applicable. This enables the Web Service Requester to view in a format that is displayable on any mobile device or browser).*

***As to Claims 12, and 32,*** *Lai anticipates method, web services program, of claims 1, and 24, wherein said optimizing further comprises extracting invocation information from said web service description, and storing said invocation information (as stated in par. 0898, lines 1-11, In this stage, a new CRM business data model (database storage) is defined and customized from the customer information extracted from existing delivery channels such as ATM channel and teller platform. The data*

extraction, one-way data synchronization using the existing middleware or messaging infrastructure is achieved by implementing web services).

***As to Claims 13, and 33, Lai anticipates method, web services program, of claims 12, and 32, further comprising processing said at least one accelerator output file by identifying web service description elements that define inputs to said web service and obtaining operation parameters based on said inputs*** (as stated in par. 0763, 0764, lines 1-3, front controller servlet initiates EPI, ECI calls, the objective is to invoke a CICS mainframe transaction ID and return the response for terminal operation parameters data).

***As to Claims 14, and 34, Lai anticipates method, web services program, of claims 13, and 33, further comprising generating input data by combining said operation parameters with said invocation information*** (as stated in par. 0777, lines 1-13, CICS resource requests from the CICS socket, web attach program and the alias program CWBA handle the conversion of the code page from ANSI to EBCDIC. The process converts input data from the request to ASCII, which may then be converted to EBCDIC. The analyzer look up the alias name, converter, user ID, and user program before passing the CICS request to the Alias program CWBA and functions as a converter, which encodes and decodes CICS data).

***As to Claims 15, and 35, Lai anticipates method, web services program, of claims 14, and 34, method of claim 14, further comprising invoking said web service by transmitting said input data to said web service*** (as stated in par. 0777, lines 1-13, Alias program then invokes the appropriate user application program to process the CICS requests. The Alias program receives the presentation logic decoded in CICS format. The Alias program sends the CICS results in encoded data that can be handled by the presentation logic such as HTML Web pages displayed on the mobile device).

***As to Claims 16, and 36, Lai anticipates method, web services program, of claims 15, and 35, further comprising receiving output data from said web service in response to said invoking step*** (as stated in par. 0777, lines 1-13, The Alias program receives the presentation logic decoded in CICS format. The Alias program sends the CICS results in encoded data that can be handled by the presentation logic such as HTML Web pages displayed on the mobile device).

***As to Claims 17, and 41, Lai anticipates method, web services program, of claims 1, and 24, wherein said at least one accelerator output file comprises code adapted for execution on said mobile device, for obtaining input data used to invoke said web service, and for invoking said web service using said input data*** (as stated in par. 0892, lines 1-15 and par. 0729, lines 4-9, J2EE-compliant applications web server provide Web and EJB containers to develop and execute Java servlets and EJBs. It also supports session, state, and connection pooling for transaction processing. JAX,

Java API for XML Pack is a bundle of XML related Java APIs to develop XML based transforming and Web service through JAXP, JAXB, JAXM, JAXR, and JAX-RPC modules to be executed on the mobile platform devices. Web service requester may start with one single coarse-grained Web service, where different services are invoked by passing parameters in the Web service call either an XML-RPC or an XML message).

***As to Claims 18, and 42, Lai anticipates method, web services program, of claims 17, and 41, wherein said creating step comprises the following substeps:***

*a) identifying web service description elements that define inputs to said web service from said web service description (as stated in par. 0301, lines 1-7, web service technologies implement service proxy, JAXR, search engines for searching registry with their search algorithm, combination of logical operators for identifying elements that define inputs to web service);*

*b) producing first instructions for generating a user interface to prompt a user for one or more of said inputs to said web service (as stated in par. 0302, lines 1-7 and par. 0310, lines 1-7 JAXR pluggable provider class is able to use any capability-specific interfaces such as ebXML provider or UDDI provider that is specific to a particular Service Registry platform and require to input to many data entry screens for creating a service record and access registry content, objects, catalogs, specifications, WSDL documents, standards and XML schema);*

*c) producing second instructions for obtaining input data associated with said one or more inputs* (as stated in par. 0303, lines 1-7, APIs or servlets are used to access their registry contents. When Java API for Registries JAXR is used, the same Java API may be used with different registry properties. Property file stores the specific registry servlet or URIs, Universal Resource Identifier, or the service endpoints URLs);

*d) identifying web service description elements that define a destination and a format for said inputs to said web service* (as stated in par. 0240, lines 1-7, web service providers use different messaging protocols and data formats, they use SOAP or ebXML messaging to exchange documents or instructions. The benefit of using SOAP or ebXML messaging is that they are able to integrate with the client or service providers' back-end systems. Documents encapsulated in XML structure within a SOAP message are transcoded into a format that can be understood by the back-end ERP or legacy systems);

*and e) producing third instructions for invoking said web service by transmitting said input data to said destination in said format* (as stated in par. 0422, lines 1-7, controller servlet are documents or instructions encapsulated in XML structure for the Integration Tier to hosts the integration components such as messaging bus, gateways such as Host Gateway for legacy mainframe systems, and security components such as Directory Server and Policy Server. The Host Gateway provides a channel to invoke applications running on legacy mainframes for appropriate web services).



***As to Claims 19, and 44, Lai anticipates method, web services program, of claims 18, and 43, wherein said creating step further comprises the substeps of:***

f) identifying web service description elements that define outputs from said web service in response to invocations of said web service and a format for said outputs from said web service description (as stated in par. 0422, lines 1-18, controller servlet are documents or instructions encapsulated in XML structure for the Integration Tier to hosts the integration components such as messaging bus, gateways such as Host Gateway for other web services. The Directory Server provides enterprise-level authentication for web services. The Policy Server stores access rights and policies that govern the access level of each web service component or system by users and by roles. These security components span two or more tiers);

*and g) producing fourth instructions for receiving output data in said format from said web service (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, messaging calls and data returned from the RPC router servlet is captured by the service proxy from the Resource Tier which hosts all data stores residing on a database server running a relational database, data warehouse, Enterprise Resource Planning (ERP) systems, and legacy mainframe applications).*

***As to Claims 20, and 45, Lai anticipates method, web services program, of claims 19, and 44, wherein said creating step further comprises the substep of:***

*h) producing fifth instructions for outputting output data received from said web service to said user (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5,*

presentation manager reformats the data and transcode into HTML, or WML using XSL if applicable. This enables the Service Requester (user) to view in a format that is displayable on any mobile device or browser).

***As to Claims 21, and 48, Lai anticipates method, web services program, of claims 17, and 35, wherein said creating step further comprises compiling instructions produced at said creating step into said code*** (as stated in par. 0872, lines 1-6, web services technologies implement cross-compiling of the source program language structure such as COBOL to a target program language structure such as Java byte-code using an intelligent language cross-compiler without changing the application program logic, structured procedures and instructions may then be transcoded into Java beans or EJBs for optimizing web services).

***As to Claims 22, and 49, Lai anticipates method, web services program, of claims 21, and 35, wherein said code represents an executable Java application*** (as stated in par. 0872, lines 1-6, web services technologies implement cross-compiling of the source program language structure such as COBOL to a target program language structure such as Java byte-code using an intelligent language cross-compiler without changing the application program logic, structured procedures and instructions may then be transcoded into Java beans or EJBs for optimizing web services).

**As to Claim 23,** *Lai anticipates method, of claim 17, further comprising processing said at least one accelerator output file by executing said code* (as stated in par. 0892, lines 1-15 and par. 0729, lines 4-9, J2EE-compliant applications web server provides Web and EJB containers to develop and execute Java servlets and EJBs. It also supports session, state, and connection pooling for transaction processing. JAX, Java API for XML Pack is a bundle of XML-related Java APIs to develop XML -based transforming and Web services through JAXP, JAXB, JAXM, JAXR, and JAX-RPC modules to be executed on the mobile platform devices. Web services requester may start with one single coarse-grained Web service, where different services are invoked by passing parameters in the Web service call either an XML-RPC or an XML message).

**As to Claim 37,** *Lai anticipates, web services program of claim 36, wherein said method further comprises transmitting at least a subset of said output data to said mobile device* (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, presentation manager reformats the data and transcode into HTML, or WML using XSL if applicable. This enables the Service Requester (user) to view in a format that is displayable on any mobile device or browser).

**As to Claim 38,** *Lai anticipates, web services program of claim 24, wherein said method further comprises receiving input data from said mobile device and invoking said web service by transmitting said input data to said web service* (as stated in par.

0967, lines 1-19, par. 0892, lines 1-15 and par. 0729, lines 4-9, client requests for the web services start with inputting data as User ID and password, are represented in SOAP messages, which are carried over HTTPS. SOAP messages may also be carried over other data transport, such as SMTP or FTP. J2EE-compliant applications web server provides Web and EJB containers to develop and execute Java servlets and EJBs. It also supports session, state, and connection pooling for transaction processing. JAX, Java API for XML Pack is a bundle of XML -related Java APIs to develop XML -based transforming and web services through JAXP, JAXB, JAXM, JAXR, and JAX-RPC modules to be executed on the mobile platform devices. web services requester may start with one single coarse-grained web service, where different services are invoked by passing parameters in the web services call either an XML -RPC or an XML message).

*As to Claim 39, Lai anticipates, web services program of claim 38, wherein said method further comprises receiving output data from said web service in response to said invoking step and transmitting said output data to said mobile device (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, presentation manager reformats the data and transcode into HTML, or WML using XSL if applicable. This enables the Service Requester (user) to view in a format that is displayable on any mobile device or browser).*

***As to Claim 40,*** *Lai anticipates, web services program of claim 39, wherein said method further comprises the step of detecting changes to said output data from said web service in response to said invoking step and transmitting said changes to said mobile device* (as stated in par. 0240, lines 1-7, web service providers use different messaging protocols and data formats, they use SOAP or ebXML messaging to exchange documents or instructions. The benefit of using SOAP or ebXML messaging is that they are able to integrate with the client or service providers' back-end systems. Documents encapsulated in XML structure within a SOAP message are transcoded into a format that can be understood by the back-end ERP or legacy systems. Thus, the integration effort may be lower and reusable for other web services, and may not require all web service providers to use the same vendor solution or to adopt a proprietary data format. This enables any changes from back-end systems to be transmitted back to Service Requester (user) to view in a format that is displayable on any mobile device or browser).

***As to Claim 43,*** *Lai anticipates, web services program of claim 42, wherein said third instructions comprise instructions for receiving said input data from said mobile device and transmitting said input data to said web service* (as stated in par. 0967, lines 1-19, par. 0892, lines 1-15 and par. 0729, lines 4-9, client requests for the web services start with inputting data as User ID and password, are represented in SOAP messages, which are carried over HTTPS. SOAP messages may also be carried over other data transport, such as SMTP or FTP. J2EE-compliant applications web server provides

Web and EJB containers to develop and execute Java servlets and EJBs. It also supports session, state, and connection pooling for transaction processing. JAX, Java API for XML Pack is a bundle of XML-related Java APIs to develop XML-based transforming and web services through JAXP, JAXB, JAXM, JAXR, and JAX-RPC modules to be executed on the mobile platform devices. web services requester may start with one single coarse-grained web service, where different services are invoked by passing parameters in the web services call either an XML -RPC or an XML message).

***As to Claim 46,*** *Lai anticipates, web services program of claim 45, wherein said fifth instructions comprise instructions for receiving said output data from said web services and for transmitting said output data to said mobile device* (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, presentation manager reformats the data and transcode into HTML, or WML using XSL if applicable. This enables the Service Requester (user) to view in a format that is displayable on any mobile device or browser).

***As to Claim 47,*** *Lai anticipates, web services program of claim 46, wherein said method further comprises the step of detecting changes to said output data from said web service in response to said invoking step and transmitting said changes to said mobile device* (as stated in par. 0240, lines 1-7, web service providers use different messaging protocols and data formats, they use SOAP or ebXML messaging to

exchange documents or instructions. The benefit of using SOAP or ebXML messaging is that they are able to integrate with the client or service providers' back-end systems. Documents encapsulated in XML structure within a SOAP message are transcoded into a format that can be understood by the back-end ERP or legacy systems. Thus, the integration effort may be lower and reusable for other web services, and may not require all web service providers to use the same vendor solution or to adopt a proprietary data format. This enables any changes from back-end systems to be transmitted back to Service Requester (user) to view in a format that is displayable on any mobile device or browser).

### ***Response to Arguments***

5. Applicant's arguments, with regards to **Claims 1, and 3-20**, filed 03/20/2008 have been fully considered but they are not persuasive.

The Examiner respectfully disagrees with Applicant's arguments, on page 11-14, regarding Claims 1, 24 and 50, as updated search resulted in new grounds of rejections.

Regarding, steps performed at mobile device, user or client or consumer from a mobile device requests web services from web services provider, hence user is also service requester. As stated in par. 0412, lines 1-10, par. 0080, lines 1-2 Service Requesters (or consumers) may be accessing the business services from a variety of mobile devices or a browser. This belongs to the consumer domain. A client may use a

phone to inquire about an account balance, where all other architecture components are part of the Service Provider domain, where the relevant Web Services components may process the balance inquiry and perform transcoding for different client mobile devices wherever necessary. A client may receive an account balance on a PDA, WAP phone, or another device based on a personalization profile. FIG. 17 illustrates an exemplary Web Services architecture showing the Consumer Domain and Service Provider Domain which is remote to the mobile device and where all the processing of requested web services is done.

Regarding the usefulness of Lai web services Examiner respectfully disagrees with Applicant's arguments on page 12-14 of Remarks. As stated in par. 1158, lines 1-12, par. 0585, lines 1-4, par. 0586, lines 1-4, par. 0587, lines 1-4 and par. 0588, lines 1-4, par. 1459, lines 1-3, This Web Services pattern may be useful, for example, for portal integration with multiple Service Providers or content providers doing processing at the back-end. Traditional point-to-point partner integration typically requires complex technical customization for each partner's back-end application infrastructure. Using Web Services technology may preferably lower the cost of partner integration because SOAP messaging may be relatively flexible and less costly to implement. It can also accommodate different data formats and platform connectivity. Business information can easily be aggregated from multiple sources to create user-stickiness for the portal service with single front-end. ProfileServlet parses the SOAP request using the JAXM/TSIK Message Provider for optimizing and resolving symbolic references for single sign-on by web service requester using mobile device. Web service description is



Art Unit: 2100

optimized with administering Multiple Servlet Engines, different configurations for different servlet engines in the server.xml file, different services and resources mapping. Vertical scaling optimizes existing computing capacity by running multiple instances of the SOAP servlet engines. Multiple servlet engines support application partitioning to preferably provide better performance and scalability);

Therefore, in view of the above reasons, Examiner maintains rejections.

#### **Action Final**

6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

***Conclusion***

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Muktesh G. Gupta whose telephone number is 571-270-5011. The examiner can normally be reached on Monday-Friday, 8:00 a.m. -5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William C. Vaughn can be reached on 571-272-3922. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Art Unit: 2100

*/John Follansbee/*

*Supervisory Patent Examiner, Art Unit 2151*